PORTAL
THE ACM DIGITAL LIBRARY

> home about > feedback > logout
US Patent & Trademark Office

Search Results

Search Results for: [leaf and branch<AND>((key<AND>((network and node and pointer and hash<AND>((tree <near> search<AND>((longest <near> match <near> prefix))))))))]
Found 8 of 96,434 searched. → Rerun within the Portal

Search within Results

> Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score **Binder**

Results 1 - 8 of 8 short listing

1 Scalable high-speed prefix matching 84%
Marcel Waldvogel , George Varghese , Jon Turner , Bernhard Plattner
ACM Transactions on Computer Systems (TOCS) November 2001
Volume 19 Issue 4

Finding the longest matching prefix from a database of keywords is an old problem with a number of applications, ranging from dictionary searches to advanced memory management to computational geometry. But perhaps today's most frequent best matching prefix lookups occur in the Internet, when forwarding packets from router to router. Internet traffic volume and link speeds are rapidly increasing; at the same time, a growing user population is increasing the size of routing tables against which p ...

2 Fast address lookups using controlled prefix expansion 82%
V. Srinivasan , G. Varghese
ACM Transactions on Computer Systems (TOCS) February 1999
Volume 17 Issue 1

Internet (IP) address lookup is a major bottleneck in high-performance routers. IP address lookup is challenging because it requires a longest matching prefix lookup. It is compounded by increasing routing table sizes, increased traffic, higher-speed links, and the migration to 128-bit IPv6 addresses. We describe how IP lookups and updates can be made faster using a set of of transformation techniques. Our main technique, controlled prefix expansion, transf ...

3 Data compression 77%
Debra A. Lelewer , Daniel S. Hirschberg
ACM Computing Surveys (CSUR) September 1987
Volume 19 Issue 3

This paper surveys a variety of data compression methods spanning almost 40 years of research, from the work of Shannon, Fano, and Huffman in the late 1940s to a technique developed in 1986. The aim of data compression is to reduce redundancy in stored or communicated data, thus increasing effective data density. Data compression has important application in the areas of file storage and distributed systems. Concepts from information theory as they relate to the goals and evaluation of data ...

4 The string B-tree 77%
Paolo Ferragina , Roberto Grossi
Journal of the ACM (JACM) March 1999
Volume 46 Issue 2

We introduce a new text-indexing data structure, the String B-Tree, that can be seen as a link between some traditional external-memory and string-matching data structures. In a short phrase, it is a combination of B-trees and Patricia tries for internal-node indices that is made more effective by adding extra pointers to speed up search and update operations. Consequently, the String B-Tree overcomes the theoretical limitations of inverted files, B-trees, prefix B-trees, s ...

5 Faster IP lookups using controlled prefix expansion 77%
V. Srinivasan , George Varghese
ACM SIGMETRICS Performance Evaluation Review , Proceedings of the joint international conference on Measurement and modeling of computer systems June 1998
Volume 26 Issue 1

6 Small forwarding tables for fast routing lookups 77%
Mikael Degermark , Andrej Brodnik , Svante Carlsson , Stephen Pink
ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication October 1997

Volume 27 Issue 4

7 Router plugins 48%

Dan Decasper , [redacted]bin Dittia , Guru Parulkar , Be[redacted]ard Plattner
IEEE/ACM Transactions on Networking (TON) February 2000
Volume 8 Issue 1

8 Data compression with finite windows 25%
E. R. Fiala , D. H. Greene
Communications of the ACM April 1989
Volume 32 Issue 4

Several methods are presented for adaptive, invertible data compression in the style of Lempel's and Ziv's first textual substitution proposal. For the first two methods, the article describes modifications of McCreight's suffix tree data structure that support cyclic maintenance of a window on the most recent source characters. A percolating update is used to keep node positions within the window, and the updating process is shown to have constant amortized cost. Other methods explore the ...

Results 1 - 8 of 8 **short listing**